A LOW-COST, IP-BASED MOBILE NETWORK EMULATOR (MNE)

Joseph P. Macker
Naval Research Laboratory
Washington, DC
And

William Chao
Naval Research Laboratory
Washington, DC
And

Jeffery W. Weston
Naval Research Laboratory
Washington, DC

ABSTRACT

The Naval Research Laboratory (NRL) Mobile Network Emulator (MNE) is a low-cost, flexible wireless mobile internetwork protocol (IP) test environment that provides flexible, dynamic topology control and manipulation for testing of both IPv4 and IPv6 dynamic network scenarios. Direct and indirect software support for network node motion modeling is supported. We describe the emulation design and various software and hardware support components. We also provide a case example of how the mobile emulation system has been applied with a set of ancillary visualization tools, motion generators, and network analysis tools. Finally, we discuss how such an emulation environment provides a valuable engineering tool supplementing more abstract simulation studies and costly, time-consuming field trials of mobile network systems and software.

BACKGROUND

The fact that mobile networking is a "hot" topic of present technology research and development hardly needs saying. Recent technical publications are inundated with reports of promising technologies and approaches for a better wireless future. One emerging technology area, mobile ad hoc networks (MANET) [1], focuses on support for autonomous operation of dynamic, wireless networks within designated localized areas of a given IP-based wireless network. This technology is seen as an important advancement for the future of network-centric military operations and has many commercial and emergency-related applications as well.

Orchestrating a live field trial of wireless mobile networking involves significant cost and logistical issues relating to mobile platforms, support personnel, network and experiment automation, antennas, and support equipment. The significant cost and logistics required to execute such a field trial can also be limiting in terms of achieving meaningful test results that exercise a practical number of mobile nodes over a significant set of test conditions within a given time. There is no argument that field trials are an important component of dynamic network testing, but while important field trial work continues, we have become increasingly convinced that flexible mobile emulation environments are needed to bridge the gap between the more abstract simulation studies and actual field trial work. We feel this is especially true in studying upper layer protocol behavior and performance under a variety of conditions.

In conceiving our work, we envisioned a mobile network emulation system that was low-cost, flexible, and controllable. One main goal was to support the emulation of node motion in several ways including: replay of captured motion trace files from live experiments, self-generation of motion base upon internal models, and the reuse of related motion generation tools typically used within simulation environments. In this way, both live field trial and simulation-based motion scenarios can drive additional emulation test experiments. This feature enables improved validation of software models and experiments under controlled conditions. Perhaps the most important application of the emulation environment is to provide a low-cost environment to execute dynamic network testing using actual software environments targeted for live wireless, mobile networks. In this sense, the emulator is a *time and cost saving* system debugging and analysis tool that supplements simulation and field studies, especially during early phases of software and protocol development.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **A Low-Cost, IP-Based Mobile Networks Emulator (MNE)**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

MOBILE NETWORK EMULATION SYSTEM DESIGN

The following section provides a brief description of our NRL-developed mobile network emulator (MNE) design. A more detailed, yet older overview can be found in [2]. Figure 1 shows a high-level overview of the MNE system approach and illustrates how it is typically applied in practice. Depicted are two separate network interfaces per emulated network device. As we shall discuss, in the most common configuration one of the physical network interfaces acts as a mobile emulation control channel while the other interface is used in emulating the actual dynamic of the network topology amongst the participating nodes. The bubble diagram in Figure 1 depicts an example multi-hop topology realized amongst the nodes.

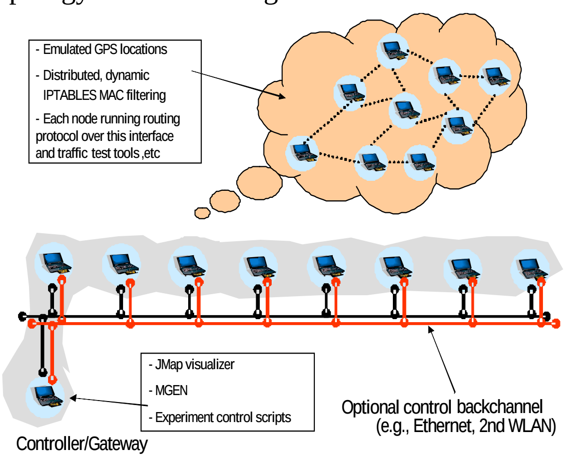

Figure 1 — A high-level view of the MNE

We emphasize that the initial emulator design goal was not to emulate the wide variety of detailed wireless communication channel conditions that are experienced in a real world scenario (e.g., hidden terminal effects, time-varying radio channel models). Rather, the main goal is low cost, controlled repeatability and the examination of dynamic topology effects at the network layer and above. More complex emulation systems are appropriate for detailed wireless channel condition emulation. Within our emulation design, by optionally using actual wireless local area network (WLAN) hardware, some of the media access (MAC) layer effects can realized in an emulator trial, but detailed performance and other issues only exposed in specific real world environments should be cross-checked with a detailed simulation model or a real environmental experiment.

FUNCTIONAL COMPONENTS

Figure 2 shows the major function components of the MNE and their relationships to other system programs. The NRL MNE is largely based upon open source operating system capabilities and uses a set of building blocks to achieve low cost and flexibility. Although it can be easily extended to work on other systems, it is presently designed to run under a variety of Linux-based distribution environments with IPTABLES network filtering capability installed. If IPTABLES is not available, we would expect the MNE approach to be easily adaptable to other dynamic network filtering utilities that can utilize MAC layer addressing.

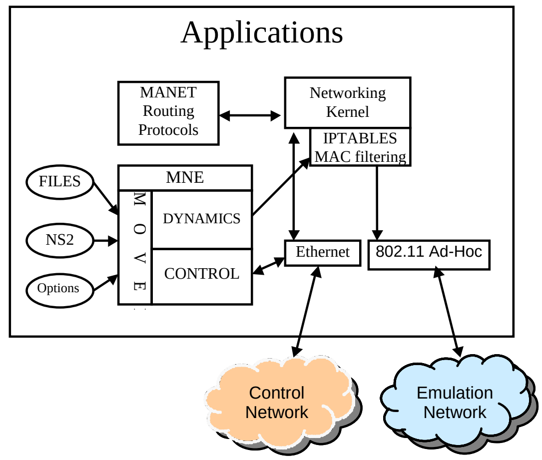

Figure 2 — MNE components

MNE Hardware Components

Within the MNE, emulation control traffic is designed to operate independently from operational network traffic sent over the controlled network topology. The node location update information can be sent via a wireless or wired interface regardless of which interface is being used for routing and user traffic. For example, we can construct a test bed with a single hard-wired interface, a single wireless interface, or both a wired and a wireless interface. In the single-wired interface environment, mobile routing protocols can be exercised under induced topology dynamics, however the actual results may suffer under congested tests since control traffic is sharing the same physical channel as potentially congested user traffic. The control channel uses a well-known multicast address and port number that is not filtered by IPTABLES and is received continuously by all nodes. Instead of using a wired Ethernet, the use of a wireless interface (such as a WLAN) may better reflect some of the MAC layer behavior and interference issues unique to wireless operation. If the single interface mode of operation is used and the associated interface becomes congested, the location information sent out by the MNE can interfere with the normal test data, resulting in a lower throughput and increased latency. Similarly, the normal test traffic can interfere with the location information, resulting in nodes receiving stale position updates for other nodes. Some priority or quality of service (QoS) packet processing of the control traffic can be done to provide probabilistic

improvements to this condition, but interference effects cannot typically be completely avoided in the single interface case. Therefore, we recommend that two network interfaces be used for general operation of the MNE, especially when network congestion tests are desired. In addition to recommendation of two interfaces, the most recommended configuration is to use both a wireless (e.g., 802.11 in the ad hoc mode) and a hard-wired Ethernet interface. Operational network data, including the routing protocol and user data, can be directed out via the wireless interface, just as it would be in a real-world scenario (albeit without potential hidden terminal effects, etc). In this recommended configuration a hard-wired interface (e.g., Ethernet) is used as the supplemental back channel for the emulation control information and status collection.

Software Components

A number of software components are used to construct the MNE and we have built upon widely available open source system tools whenever possible. We have developed some unique software to control the emulation motion and to distribute the control data. We also base our motion interface upon global positioning system (GPS) reference location data so that we can apply the same test tools, methods, and tracing tools used in the real world test with actual working GPS components.

IPTABLES Filtering

In the initial design, a simple propagation range model is used to emulate a nominal wireless link range, and a dynamic method for blocking packets from particular nodes out of range is used. The PREROUTING filtering chain function of IPTABLES is a dynamic way to achieve this in a low cost open design manner. IPTABLES (IP6TABLES for IPv6) is standard system software that comes with a variety of open source operating systems (e.g., Linux) to filter network packets on a given network interface. By inserting and deleting entries in the PREROUTING chain of the IPTABLES one can emulate logical wireless connectivity or topological dynamics. Although supportable, filtering by IP address was not chosen to perform the blocking and unblocking function because the nature of MANET routing and generic IP routing is to forward IP traffic on the behalf of other source nodes. Blocking a particular IP source address is incorrect behavior and would disrupt the emulation of the IP forwarding capability. A second point is that nodes cannot be blocked via IP addresses if they do not have an IP address, which may often be true, as when a node is performing broadcast requests or discovery services prior to obtaining a routable source address.

The uniform range model used in the initial design of link activation/deactivation is simple and allows for controlled experiments of scenario-driven dynamic network topologies. In recent updates to the link range mode, we have introduced the notion of statistical quality variation to the range model characteristics. Another enhancement is a simplified emulation of wireless blockage within the emulated area that can be used to crudely emulate real-world wireless obstacles such as buildings, towers, and hills. At present, we are discussing and considering enhancements to link modeling, but we also wish to preserve the low complexity, low overhead nature of the present MNE design.

Multicast-based Emulation Control Channel

The nodes in the MNE environment need to know the location of every other node in the test, as they use such information to determine dynamic connectivity effects in real time. Since the updating of this information needs to be sent to all participating nodes, multicasting the data is a logical design choice. Since basic IP multicast transport is an efficient but best-effort group delivery mechanism, we use a reliable multicast transport mechanism to ensure more robust reception of the control channel data. A group-based reliable multicast program called *mdpchat* [3] was used as the framework for the MNE multicast control, namely the multicast of location information for MAC filtering. Using *mdpchat* as our base design, we have the ability to reliably and efficiently multicast location and experimental control information to all nodes over the backchannel with a reasonably high degree of confidence.

Node Motion Emulation

A good portion of the MNE design involves supporting the distributed generation or replay of motion models to control an experimental scenario. The motion emulation portion of the MNE involves the following three distinct software components integrated into one software program.

- **Movement** module
- **Dynamics** module
- **Control** module

The **movement module** provides the location information — typically represented in terms of longitude and latitude coordinates. The **dynamics module** determines node ranges and potential wireless blockages, and performs dynamic filtering. The **control module** provides a communications backchannel for sharing dynamic location information between nodes.

The MNE supports a wealthy set of motion generation models. The supported option models are “file,” “random,”

"master/slave," "waypoints," "ns2," "static," "line," and "circle." In **file** mode, the movement module plays back a prerecorded motion file for a set of mobile nodes that were logged by the NRL gpsLogger program [4] during a previous live field test. In **random** mode, the movement module applies a variant of a random vector model for motion [5]. In the **master/slave** mode, a designated master node sends out the locations on behalf of all slave nodes. The slaves listen to these messages not only to see where other nodes are, but also to get updates on their own current location. The **waypoints** model supports motion testing where nodes travel to pre-determined waypoints. In the **ns2** motion mode, NS2 network simulator motion files can be used as input. The NS2 file is a particular movement description file used by the NS2 network simulation environment [6]. Various motion generators, such as the Bonn University Java-based mobility generator [7] and Carnegie Mellon University Monarch mobility generator [5] can also be used to generate these NS2 movement files. The collection of available tools supports a variety of different motion models including; random vectors, grid motion, and clustered motion. The **static** motion mode places a node at a particular fixed location.

OTHER RELATED TOOLS

During the course of its design, the MNE was integrated and tested with several other mobile network testing support tools. Those tools are not the focus of this document, but they are mentioned because they support a common software test environment for both emulation and live field testing. The Multi-GENerator (MGEN) is a network traffic test tools and is used to generate traffic and to log received traffic for performing analyses of mobile network performance [4]. Java-based Mobile Visualizer and Mapping Application (JMap) [4] is a mobile network visualization tool developed at NRL and used to support a wide variety of testing and experimental applications. JMap supports dynamic display of dynamic network node positions and associated routing links between nodes in an experiment. A similar, but more flexible visualization tool, Scripted Display Tool (SDT) in combination with the Scripted Display Tool Parser (sdtParser) [4], both written in C++ by NRL, can also be used to display node position and routing links.

Overall, the MNE design is unobtrusive to the types of IP and network protocols used during testing and it supports both IPv4 and IPv6 protocol testing. At present, NRL has used the tool in testing a number of MANET protocols including variants of the Optimized Link State Routing (OLSR) [8] protocol by INRIA, an OLSR implementation created by NRL [4], a version of OLSR for IPv6 [4], and variants of the Ad-hoc On-demand Distance Vector (AODV) [9] protocol.

EXAMPLE USE AND RESULTS

In this example, the output from a real world MANET test with a captured node motion model and known traffic pattern is used to create an emulation scenario. The node motion and traffic models are replayed in the emulation environment using the same testing hardware and software. A sample live test was conducted at NRL using a prototype MANET routing protocol implementation; in this case, a version of the OLSR protocol implementation was used. In the live test, ten nodes were deployed with one remaining in a fixed location (Xcom), while the other nine nodes moved every four minutes from one waypoint to the next. A set of buildings and counter rotating motion at reasonable ranges produced continuous topology updates. A network traffic pattern consisting of three phases was used. The first phase involved the nine mobile nodes each sending 10 kbps to the stationary node (Xcom) for 10 minutes. Xcom also sent 10 kbps of data to itself for visualization purposes, but this data is sent across the loopback, rather than the wireless interface. During the second phase, the rate for each node was increased to 50 kbps for 10 minutes. The rate was again increased to 100 kbps per node in the third phase, a traffic level known to produce congestion and loss. The motion waypoint rotation period for each phase remained at four minutes throughout the test. Figure 3 shows a JMap snapshot during the testing of the nodes and OLSR routing links. The links depict the routes between the peer nodes from the Xcom perspective. Xcom is designated as "1:100" in the Figure 3.

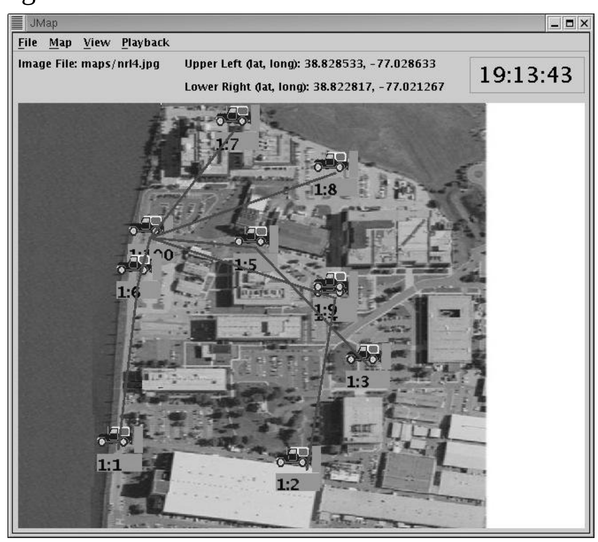

Figure 3 — JMap visualization from example test

The MNE test results are compared with those of the live test in the following figures. The data rate performance comparison is shown in Figures 4 and 5. The throughput rate in Figure 4 is the real-world throughput

and the throughput rate in Figure 5 is the emulated movement. The link range for the MNE run was set to be nominally 300 meters and standards 802.11b wireless cards were used in both tests. The GPS location that was logged during the live test is used as input to the emulator's file playback.

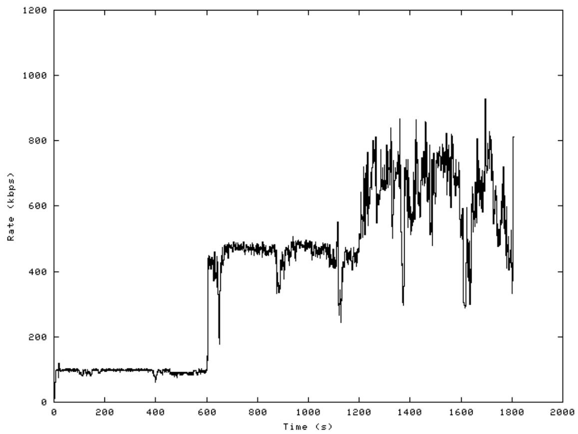

Figure 4 — Live test data throughput rate

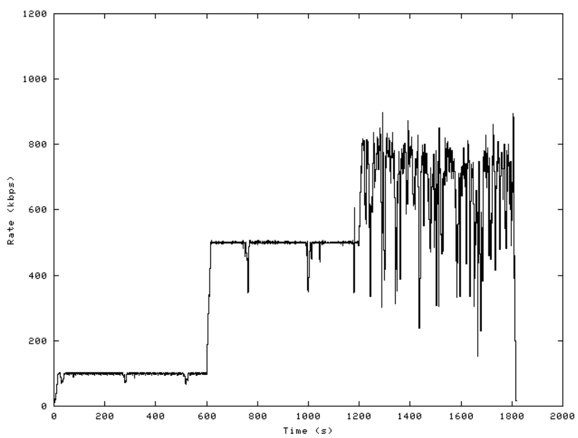

Figure 5 — Emulated test data throughput rate

OTHER SUPPORTED EXPERIMENTS

In this section, we illustrate some additional mobile network experiments that have presently been supported by the MNE. In a set of experiments, we have used the MNE to examine MANET overall performance enhancements when different types of enhanced forms of packet forwarding were applied at the network layer. One experiment performed examined the effect of prioritizing user data packets under congested conditions. In this experiment, aggregate mobile node source traffic of 1.4 Mbps is relayed within a 10-node multi-hop MANET network running an implementation of the OLSR protocol. This data is divided into two flows of 700 Kbps each – one with Quality of Service (QoS) markings, one without. The routing control messages were also given the highest priority, since routing is vital to user data throughput. The results, shown in Figure 6, demonstrate the effect of adding QoS to data flows.

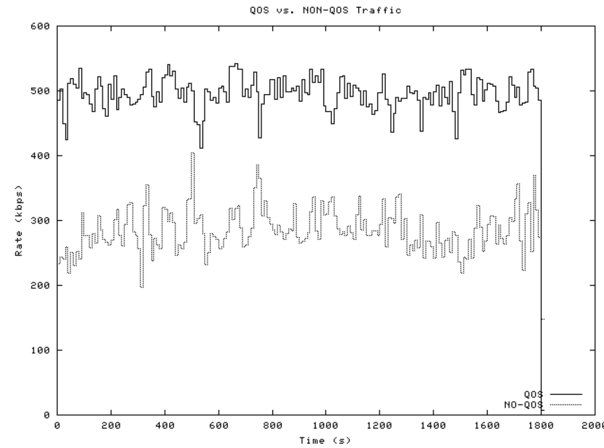

Figure 6 — Data throughput rate with and without QoS

In another set of experiments, the use of a Dynamic Host Configuration Protocol (DHCP) server [10] approach to support node configuration within a multi-hop MANET stub network was demonstrated and examined. This was made possible by using a managed DHCP server configuration at the MANET gateway node and by supporting DHCP relay agent [11] functionality on each mobile node within the MANET network. This was done in both IPv4 and IPv6. Figure 7 depicts how the MNE is used to create an environment to test the auto-configuration capability in a mobile scenario. When a new node Y comes in contact with a node X on the edge of the MANET, the DHCP offer-and-request protocol is relayed to provide auto-configuration of the new node. These experiments demonstrate the flexibility of the MNE and its ability to examine other interesting network problems supplementing mobile routing protocol examination.

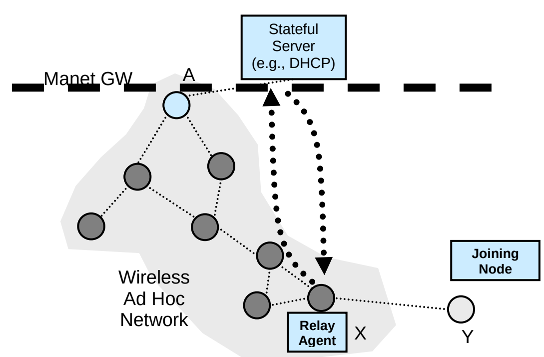

Figure 7 — MNE use for MANET and DHCP relay Experiment

POSSIBLE FUTURE WORK

At present, the MNE is a simple, low-cost mobile emulation toolkit that does not require significant amounts of processing or sophisticated propagation models to

produce useful experimental test results. It has proved valuable as a protocol and system debugging tool for testing prototype systems designed for dynamic, mobile network operation. As has been mentioned, the link dynamics model is simple in nature and is applicable mainly for higher layer protocol testing (i.e., layer 3 and above). At present, the tool is flexible in supporting a wide variety of motion models and IP layer routing protocols and has recently been used in testing the performance of MANET multicast routing as well.

The authors are interested in enhancing the MNE in the future to better support the testing and automation of experimentation involving not just mobile routing, but also distributed auto-addressing and configuration of nodes. The MNE can be enhanced to support more dynamic user traffic scenarios as mobile nodes change roles within a network throughout an experiment.

SUMMARY AND CONCLUSIONS

We have described the design and features of an NRL-developed, low-cost Mobile Network Emulator (MNE). While other emulation work has been done, we feel the MNE is a valuable addition as a dynamic network testing toolkit for several reasons. First, it supports multiple mobility models and interface mechanisms including; playback of input traces from live experiments and external input from other tools (e.g., simulation tools and third party motion tools). Second, the MNE is simple to deploy in a laboratory environment by building on standard, low-cost operating system functions for packet filtering. Third, while remaining low cost and easily integrated, the MNE provides a sophisticated and extensible backchannel control mechanism to eliminate performance issues in distributed network testing. Fourth, the MNE realizes the ability to run repeated, automated mobile networking experiments within a small laboratory space using the same automated test tools and software systems used in live field testing.

In addition to describing the MNE design, we presented a case example involving playback of motion and traffic patterns from a field experiment while testing a prototype MANET routing protocol. We also briefly described NRL–developed data analysis and visualization software tools typically used in both field trials and within the MNE to perform experiments. In conclusion, the MNE is an inexpensive mobile network emulation tool that does not replace, but supplements simulation and actual real world experimental work. The MNE easily runs on typical laptop computers with a wide variety of standard network cards providing a low-cost, flexible bench top mobile network laboratory. It can also be easily configured to produce automated tests and data collection, reducing the need for monitoring and interaction during lengthy testing periods.

REFERENCES

[1] S. Corson and J. P. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Consideration,” IETF Request for Comments: 2501, January 1999.

[2] W. Chao, J. Macker, J. Weston, “NRL Mobile Network Emulator”, NRL Formal Report 5523—03-10, 054, Feb 6, 2003.

[3] J.P. Macker and R.B. Adamson, “The Multicast Dissemination Protocol Toolkit,” Proceedings IEEE MILCOM 99, Oct. 31-Nov. 3, 1999, paper 21.1.

[4] Protean Research Group Software Tools website, Naval Research Laboratory, http://pf.itd.nrl.navy.mil/.

[5] D.A. Maltz, J. Broch, and D.B. Johnson, “Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed,” Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon Univ, March 1999.

[6] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, “Advances in Network Simulation,” *IEEE Computer* 33(5), 59-67, May 2000.

[7] Bonn University Mobility Generator, 2002, http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/.

[8] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized Link State Routing Protocol,” Proceedings IEEE National Multi-Topic Conference (INMIC) 2001, December 28-31,2001.

[9] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordström, C. Tschudin, "A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations", WCNC 2002.

[10] R. Droms, “Dynamic Host Configuration Protocol”, Network Working Group RFC 2131, March 1997.

[11] M. Patrick, “DHCP Relay Agent Information Option”, Network Working Group RFC 3046, January 2001.